

## ***Examples of questions and replies from AskProfi.com***

<b>No.</b>	<b>Your question</b>	<b>Reply from Ask Profi</b>
1.	I want to import application data from data file into Oracle database. What is the fastest, most flexible and easy way to do this?	We would advice you to use XML data files for your import. You can use XML parser for Oracle (i.e. DOM or SAX based parser) to parse your XML file. The SAX based parser is highly recommended for large XML data transfer volumes (from 10,000 and more records imported to database). Parsing refers to the process by which programming data input is broken into smaller, more distinct chunks of information (in our case, XML tags) that can be more easily interpreted and acted upon. If you need programming code in PL/SQL or Java, we can provide it to you on request.
2.	We have a distributed software application which we would like to see more flexible and configurable based on various factors. What would be the best solution for such problem?	We would propose to divide your system into autonomous components (for example, intelligent agents) that can operate independently of each other, achieving a goal of easy-to-configure system.
3.	We are looking for a high-performance method of indexing spatial data. We tried to use conventional spatial index structures such as R-trees to do this, but did not achieve acceptable performance in our software application. Do you have any ideas of how to solve it?	Non-uniformity in data extents is a general characteristic of spatial data. Indexing such non-uniform data using conventional spatial index structures such as R-trees is inefficient for two reasons: the non-uniformity increases the likelihood of overlapping index entries and clustering of non-uniform data is likely to index more dead space than clustering of uniform data. To make the impact of these anomalies more "useful", one can invent a new scheme that promotes data objects to higher levels in tree-based index structures; these objects then fulfill different functions based on the positional context in the index tree.